Ruby master - Bug #13553

Improve performance in where push the element into non shared Array object

05/10/2017 02:38 AM - watson1978 (Shizuo Fujita)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:		**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

rb_ary_modify() has the codes for shared Array object.
In here, it has condition branch for shared / non shared Array object and
it can use rb_ary_modify_check() which is smaller function than rb_ary_modify()
for non shared object.

rb_ary_modify_check() will be expand as inline function.
If it will compile with GCC, Array#<< will be faster around 8%.

Clang 802.0.42

Before

```
Calculating -------------------------------------
            Array#<<      9.353M (± 1.7%) i/s -     46.787M in   5.004123s
          Array#push      7.702M (± 1.1%) i/s -     38.577M in   5.009338s
     Array#values_at      6.133M (± 1.9%) i/s -     30.699M in   5.007772s
```

After

```
Calculating -------------------------------------
            Array#<<      9.458M (± 2.0%) i/s -     47.357M in   5.009069s
          Array#push      7.921M (± 1.8%) i/s -     39.665M in   5.009151s
     Array#values_at      6.377M (± 2.3%) i/s -     31.881M in   5.001888s
```

Result

```
Array#<<        -> 1.2% faster
Array#push      -> 2.8% faster
Array#values_at -> 3.9% faster
```

GCC 7.1.0

Before

```
Calculating -------------------------------------
            Array#<<     10.497M (± 1.1%) i/s -     52.665M in   5.017601s
          Array#push      8.527M (± 1.6%) i/s -     42.777M in   5.018003s
     Array#values_at      7.621M (± 1.7%) i/s -     38.152M in   5.007910s
```

After

```
Calculating -------------------------------------
            Array#<<     11.403M (± 1.3%) i/s -     57.028M in   5.001849s
          Array#push      8.924M (± 1.3%) i/s -     44.609M in   4.999940s
     Array#values_at      8.291M (± 1.4%) i/s -     41.487M in   5.004727s
```

Result

```
Array#<<        -> 8.3% faster
```

```
Array#push      -> 4.3% faster
Array#values_at -> 8.7% faster
```

Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|

  x.report "Array#<<" do |i|
    i.times { [1,2] << 3 }
  end

  x.report "Array#push" do |i|
    i.times { [1,2].push(3) }
  end

  x.report "Array#values_at" do |i|
    ary = [1, 2, 3, 4, 5]
    i.times { ary.values_at(0, 2, 4) }
  end

end
```

Patch

https://github.com/ruby/ruby/pull/1609

Associated revisions

Revision 044257c0 - 05/24/2017 06:57 AM - watson1978 (Shizuo Fujita)

Improve performance in where push the element into non shared Array object

- array.c (ary_ensure_room_for_push): use rb_ary_modify_check() instead of rb_ary_modify() to check whether the object can be modified for non shared Array object. rb_ary_modify() has the codes for shared Array object too. In here, it has condition branch for shared / non shared Array object and it can use rb_ary_modify_check() which is smaller function than rb_ary_modify() for non shared object.

  rb_ary_modify_check() will be expand as inline function.
  If it will compile with GCC, Array#<< will be faster around 8%.

  [ruby-core:81082] [Bug #13553] [Fix GH-1609]

Clang 802.0.42

Before

```
        Array#<<      9.353M (± 1.7%) i/s -     46.787M in   5.004123s
      Array#push      7.702M (± 1.1%) i/s -     38.577M in   5.009338s
 Array#values_at      6.133M (± 1.9%) i/s -     30.699M in   5.007772s
```

After

```
        Array#<<      9.458M (± 2.0%) i/s -     47.357M in   5.009069s
      Array#push      7.921M (± 1.8%) i/s -     39.665M in   5.009151s
 Array#values_at      6.377M (± 2.3%) i/s -     31.881M in   5.001888s
```

Result

Array#<<   -> 1.2% faster
Array#push   -> 2.8% faster
Array#values_at -> 3.9% faster

GCC 7.1.0

Before

```
        Array#<<     10.497M (± 1.1%) i/s -     52.665M in   5.017601s
      Array#push      8.527M (± 1.6%) i/s -     42.777M in   5.018003s
 Array#values_at      7.621M (± 1.7%) i/s -     38.152M in   5.007910s
```

After

```
        Array#<<     11.403M (± 1.3%) i/s -     57.028M in   5.001849s
      Array#push      8.924M (± 1.3%) i/s -     44.609M in   4.999940s
 Array#values_at      8.291M (± 1.4%) i/s -     41.487M in   5.004727s
```

Result

Array#<< -> 8.3% faster
Array#push -> 4.3% faster
Array#values_at -> 8.7% faster

Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|

x.report "Array#<<" do |i|
i.times { [1,2] << 3 }
end

x.report "Array#push" do |i|
i.times { [1,2].push(3) }
end

x.report "Array#values_at" do |i|
ary = [1, 2, 3, 4, 5]
i.times { ary.values_at(0, 2, 4) }
end

end
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@58867 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 58867 - 05/24/2017 06:57 AM - watson1978 (Shizuo Fujita)

Improve performance in where push the element into non shared Array object

- array.c (ary_ensure_room_for_push): use rb_ary_modify_check() instead of
  rb_ary_modify() to check whether the object can be modified for non shared
  Array object. rb_ary_modify() has the codes for shared Array object too.
  In here, it has condition branch for shared / non shared Array object and
  it can use rb_ary_modify_check() which is smaller function than
  rb_ary_modify() for non shared object.

  rb_ary_modify_check() will be expand as inline function.
  If it will compile with GCC, Array#<< will be faster around 8%.

  [ruby-core:81082] [Bug #13553] [Fix GH-1609]

Clang 802.0.42

Before

```
        Array#<<      9.353M (± 1.7%) i/s -     46.787M in   5.004123s
      Array#push      7.702M (± 1.1%) i/s -     38.577M in   5.009338s
 Array#values_at      6.133M (± 1.9%) i/s -     30.699M in   5.007772s
```

After

```
        Array#<<      9.458M (± 2.0%) i/s -     47.357M in   5.009069s
      Array#push      7.921M (± 1.8%) i/s -     39.665M in   5.009151s
 Array#values_at      6.377M (± 2.3%) i/s -     31.881M in   5.001888s
```

Result

Array#<<   -> 1.2% faster
Array#push   -> 2.8% faster
Array#values_at -> 3.9% faster

GCC 7.1.0

Before

```
        Array#<<     10.497M (± 1.1%) i/s -     52.665M in   5.017601s
      Array#push      8.527M (± 1.6%) i/s -     42.777M in   5.018003s
 Array#values_at      7.621M (± 1.7%) i/s -     38.152M in   5.007910s
```

After

```
        Array#<<     11.403M (± 1.3%) i/s -     57.028M in   5.001849s
      Array#push      8.924M (± 1.3%) i/s -     44.609M in   4.999940s
 Array#values_at      8.291M (± 1.4%) i/s -     41.487M in   5.004727s
```

Result

Array#<<   -> 8.3% faster
Array#push   -> 4.3% faster
Array#values_at -> 8.7% faster

Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|

x.report "Array#<<" do |i|
i.times { [1,2] << 3 }
end

x.report "Array#push" do |i|
i.times { [1,2].push(3) }
end

x.report "Array#values_at" do |i|
ary = [1, 2, 3, 4, 5]
i.times { ary.values_at(0, 2, 4) }
end

end
```

Revision 58867 - 05/24/2017 06:57 AM - watson1978 (Shizuo Fujita)

Improve performance in where push the element into non shared Array object

- array.c (ary_ensure_room_for_push): use rb_ary_modify_check() instead of rb_ary_modify() to check whether the object can be modified for non shared Array object. rb_ary_modify() has the codes for shared Array object too. In here, it has condition branch for shared / non shared Array object and it can use rb_ary_modify_check() which is smaller function than rb_ary_modify() for non shared object.

  rb_ary_modify_check() will be expand as inline function.
  If it will compile with GCC, Array#<< will be faster around 8%.

  [ruby-core:81082] [Bug #13553] [Fix GH-1609]

Clang 802.0.42

Before

```
        Array#<<      9.353M (± 1.7%) i/s -     46.787M in   5.004123s
      Array#push      7.702M (± 1.1%) i/s -     38.577M in   5.009338s
 Array#values_at      6.133M (± 1.9%) i/s -     30.699M in   5.007772s
```

After

```
        Array#<<      9.458M (± 2.0%) i/s -     47.357M in   5.009069s
      Array#push      7.921M (± 1.8%) i/s -     39.665M in   5.009151s
 Array#values_at      6.377M (± 2.3%) i/s -     31.881M in   5.001888s
```

Result

Array#<< -> 1.2% faster
Array#push -> 2.8% faster
Array#values_at -> 3.9% faster

GCC 7.1.0

Before

```
        Array#<<     10.497M (± 1.1%) i/s -     52.665M in   5.017601s
      Array#push      8.527M (± 1.6%) i/s -     42.777M in   5.018003s
 Array#values_at      7.621M (± 1.7%) i/s -     38.152M in   5.007910s
```

After

```
        Array#<<     11.403M (± 1.3%) i/s -     57.028M in   5.001849s
      Array#push      8.924M (± 1.3%) i/s -     44.609M in   4.999940s
 Array#values_at      8.291M (± 1.4%) i/s -     41.487M in   5.004727s
```

Result

Array#<< -> 8.3% faster
Array#push -> 4.3% faster
Array#values_at -> 8.7% faster

Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|

x.report "Array#<<" do |i|
i.times { [1,2] << 3 }
end

x.report "Array#push" do |i|
i.times { [1,2].push(3) }
end

x.report "Array#values_at" do |i|
ary = [1, 2, 3, 4, 5]
i.times { ary.values_at(0, 2, 4) }
end

end
```

Revision 58867 - 05/24/2017 06:57 AM - watson1978 (Shizuo Fujita)

Improve performance in where push the element into non shared Array object

- array.c (ary_ensure_room_for_push): use rb_ary_modify_check() instead of rb_ary_modify() to check whether the object can be modified for non shared Array object. rb_ary_modify() has the codes for shared Array object too. In here, it has condition branch for shared / non shared Array object and it can use rb_ary_modify_check() which is smaller function than rb_ary_modify() for non shared object.

  rb_ary_modify_check() will be expand as inline function.
  If it will compile with GCC, Array#<< will be faster around 8%.

  [ruby-core:81082] [Bug #13553] [Fix GH-1609]

Clang 802.0.42

Before

```
        Array#<<      9.353M (± 1.7%) i/s -     46.787M in   5.004123s
```

```
     Array#push      7.702M (± 1.1%) i/s -     38.577M in   5.009338s
Array#values_at      6.133M (± 1.9%) i/s -     30.699M in   5.007772s
```

After

```
       Array#<<      9.458M (± 2.0%) i/s -     47.357M in   5.009069s
     Array#push      7.921M (± 1.8%) i/s -     39.665M in   5.009151s
Array#values_at      6.377M (± 2.3%) i/s -     31.881M in   5.001888s
```

Result

Array#<< -> 1.2% faster
Array#push -> 2.8% faster
Array#values_at -> 3.9% faster

GCC 7.1.0

Before

```
       Array#<<     10.497M (± 1.1%) i/s -     52.665M in   5.017601s
     Array#push      8.527M (± 1.6%) i/s -     42.777M in   5.018003s
Array#values_at      7.621M (± 1.7%) i/s -     38.152M in   5.007910s
```

After

```
       Array#<<     11.403M (± 1.3%) i/s -     57.028M in   5.001849s
     Array#push      8.924M (± 1.3%) i/s -     44.609M in   4.999940s
Array#values_at      8.291M (± 1.4%) i/s -     41.487M in   5.004727s
```

Result

Array#<< -> 8.3% faster
Array#push -> 4.3% faster
Array#values_at -> 8.7% faster

Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|

x.report "Array#<<" do |i|
i.times { [1,2] << 3 }
end

x.report "Array#push" do |i|
i.times { [1,2].push(3) }
end

x.report "Array#values_at" do |i|
ary = [1, 2, 3, 4, 5]
i.times { ary.values_at(0, 2, 4) }
end

end
```

History

#1 - 05/24/2017 06:57 AM - watson1978 (Shizuo Fujita)

- Status changed from Open to Closed

Applied in changeset trunk|r58867.

Improve performance in where push the element into non shared Array object

- array.c (ary_ensure_room_for_push): use rb_ary_modify_check() instead of rb_ary_modify() to check whether the object can be modified for non shared Array object. rb_ary_modify() has the codes for shared Array object too. In here, it has condition branch for shared / non shared Array object and it can use rb_ary_modify_check() which is smaller function than rb_ary_modify() for non shared object.

rb_ary_modify_check() will be expand as inline function.
If it will compile with GCC, Array#<< will be faster around 8%.

[ruby-core:81082] [Bug #13553] [Fix GH-1609]

Clang 802.0.42

Before

```
        Array#<<      9.353M (± 1.7%) i/s -     46.787M in   5.004123s
      Array#push      7.702M (± 1.1%) i/s -     38.577M in   5.009338s
 Array#values_at      6.133M (± 1.9%) i/s -     30.699M in   5.007772s
```

After

```
        Array#<<      9.458M (± 2.0%) i/s -     47.357M in   5.009069s
      Array#push      7.921M (± 1.8%) i/s -     39.665M in   5.009151s
 Array#values_at      6.377M (± 2.3%) i/s -     31.881M in   5.001888s
```

Result

Array#<<   -> 1.2% faster
Array#push   -> 2.8% faster
Array#values_at -> 3.9% faster

GCC 7.1.0

Before

```
        Array#<<     10.497M (± 1.1%) i/s -     52.665M in   5.017601s
      Array#push      8.527M (± 1.6%) i/s -     42.777M in   5.018003s
 Array#values_at      7.621M (± 1.7%) i/s -     38.152M in   5.007910s
```

After

```
        Array#<<     11.403M (± 1.3%) i/s -     57.028M in   5.001849s
      Array#push      8.924M (± 1.3%) i/s -     44.609M in   4.999940s
 Array#values_at      8.291M (± 1.4%) i/s -     41.487M in   5.004727s
```

Result

Array#<<   -> 8.3% faster
Array#push   -> 4.3% faster
Array#values_at -> 8.7% faster

Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|

x.report "Array#<<" do |i|
i.times { [1,2] << 3 }
end

x.report "Array#push" do |i|
i.times { [1,2].push(3) }
end

x.report "Array#values_at" do |i|
ary = [1, 2, 3, 4, 5]
i.times { ary.values_at(0, 2, 4) }
end

end
```